AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. - 75. (canceled)

76. (currently amended): The method of claim 81, wherein <u>in step (a)</u>

a flip-flop is marked, if any of its ~~corresponding~~ latch interface constraints belong to the said unsatisfiable core<u>, and</u>

a flip-flop is marked, if its initial state value constraint belongs to the said unsatisfiable core.

77. (currently amended): The method of claim 81, wherein<u> in step (a)</u>

a flip-flop is marked, if any of its ~~corresponding~~ latch interface constraints belong to the said unsatisfiable core<u>, and</u>

a flip-flop is ~~not marked~~ <u>unmarked</u> if only its initial state value <u>constraint</u> belongs to the said unsatisfiable core~~, in which case a constraint for this initial input is added to the abstract model without adding the flip-flop~~.

78. (currently amended): The method of claim 81, wherein <u>during the bounded model checking</u>

a ~~lazy~~ modified initial value constraint is used for flip-flop, ~~instead of an eager 1-literal constraint denoting an initial value of a flip-flop,~~ with an initial value constraint m ~~is~~ being replaced by (m+y)(m+!(not(y)), where y is a fresh variable not used in rest of the satisfiability formula. ~~introduced as a dummy variable to delay propagation of effect of said initial value constraint in the satisfiability solver,~~

~~said lazy constraint making it less likely that a flip-flop is marked due to its initial value constraint being present in the said unsatisfiable core, thereby leading to a smaller abstract model.~~

79. (currently amended): The method of claim 81, wherein during the bounded model checking

a ~~lazy~~ modified constraint is used for an external constraint node, ~~instead of an eager 1-literal constraint denoting an environmental constraint, wherein~~ with an environmental constraint (m) ~~is~~ being replaced by (m+y)(m+!not(y)), where y is a fresh variable not used in rest of the satisfiability formula ~~introduced as a dummy variable to delay the propagation of the effect of the said environmental constraint in the satisfiability solver.~~

~~the said lazy constraint making it less likely that an external constraint node is marked due to its environmental constraint being present in the said unsatisfiable core, thereby leading to a smaller abstract model~~ .

80. (canceled):

81. (currently amended): A computer implemented method for ~~generating an abstract model for~~ verification of a given correctness property on a sequential circuit design ~~of an electronic circuit for verification of a given correctness property, comprising the steps of:~~ by using a satisfiability-based check for bounded model checking, comprising:

~~(a) unrolling the sequential design of the electronic circuit with the given correctness property, and environmental constraints up to some depth *k*~~

~~(b) solving the resulting satisfiability problem to determine whether the given correctness property is violated~~

~~(c) deriving an unsatisfiable core from the proof of unsatisfiability when the given correctness property is not violated, where an unsatisfiable core is a subset of the constraints that is guaranteed to be sufficient for showing that the problem is unsatisfiable~~

(a) if the given correctness property is proved correct at a depth *k*, marking only flip-flops and external constraint nodes in the circuit design are marked, based on whether their constraints appear in an unsatisfiable core generated from the proof of unsatisfiability by the satisfiability solver; otherwise, terminating verification,

(b) deriving an abstract model consisting of combinational fanin cones of the marked flip-flops and external constraint nodes,

(c) checking the given correctness property on the derived abstract model, and

- if the derived abstract model is proved correct, terminating the verification and deeming the circuit design to be correct.

- otherwise, increasing the depth of unrolling k and repeating steps (a)-(c) until either the given correctness property is violated at some depth or the circuit design is determined to be correct.

~~(d) using the said unsatisfiable core to derive an abstract model of the sequential design of the electronic circuit for further verification of the sequential design, wherein,~~

~~a subset of flip-flops and external constraint nodes in the sequential design of the electronic circuit are marked based on certain related constraints being present in said unsatisfiable core~~

~~the said abstract model consisting of complete combinational fanin cones of only the marked flip-flops and the marked external constraint nodes, such that outputs from the unmarked flip-flops are regarded as pseudo-primary inputs, and~~

~~said abstract model providing a benefit that it is guaranteed to preserve the correctness of the given correctness property up to the said finite depth k.~~

82. (currently amended): A computer implemented method for ~~generating an abstract model for a sequential design of an electronic circuit for~~ verification of a given correctness property on a sequential circuit design, comprising the steps:

~~a) unrolling the sequential design of the electronic circuit to depth k (where $k = kmin$ at the start)~~

~~b) deriving an abstract model of the sequential design of the electronic circuit at depth k, that preserves correctness of the given correctness property up to depth k, when the given correctness property is not violated~~

(a) if the property is violated at depth k using a satisfiability-based bounded model checking, reporting a counterexample is reported if it is real and terminating the verification,

(b) if the property is proved correct at depth k using a satisfiability-based bounded model checking, deriving an abstract model A_n from the proof of unsatisfiability generated by the satisfiability solver,

(c) increasing the depth of unrolling k in bounded model checking,

(d) repeating steps (a - c), constituting an *inner loop* comprising the bounded model checking iterations, until either there is no change in the size of the derived abstract model as the depth of unrolling k is increased, or some limit *kmax* on k is reached,

(e) repeating steps (a - d) by using bounded model checking on the derived abstract model A_n to derive a new abstract model A_{n+1}, these steps constituting an outer *loop* comprising the abstraction iterations, until either there is no change in the size of the derived abstract model or some limit on number of abstraction iterations is reached,

(f) checking the given correctness property on the derived abstract model, and if the derived abstract model is proved correct, deeming the circuit design to be correct; otherwise, deeming the verification to be inconclusive.

~~c) iteratively increasing *k* up to some limit *kmax*, and repeating above steps (a - c) until either the size of the said abstract model of the sequential design of the electronic circuit remains unchanged over a predetermined number of consecutive depths, or the limit *kmax* is reached.~~

83. (currently amended): The method of Claim ~~82~~ 81, where the size of the derived abstract model corresponds to the number of flip-flops in the derived abstract model.

84. (canceled)